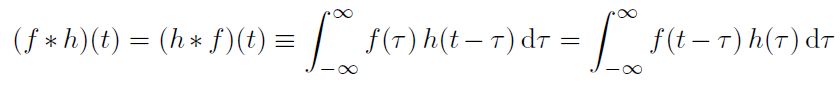
**2. Konvoluce, korelace, Fourierova transformace pro obrazy. Vzorkovací věta, rekonstrukce obrazu, interpolace obrazu, např. při geometrických transformacích.**

**Konvoluce, definice, spojitý případ**

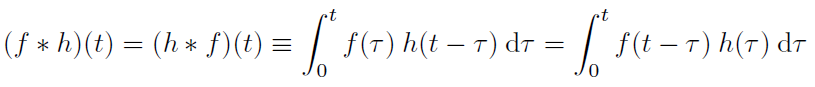
Konvoluce (ve funkcionální analýze) je operace na dvou funkcích *f* a *h*, která vytvoří třetí funkci (*f \* h*), která se používá jako modifikace jedné ze vstupních funkcí.

Konvoluce je integrál “míchající” hodnoty dvou funkcí, a to funkce *h*(*t*), která je posouvána a překrývá se s funkcí *f*(*t*) nebo obráceně.

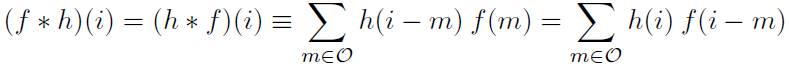
Uvažujme nejdříve spojitý případ s obecnými nekonečnými mezemi



Meze integrálu můžeme omezit na interval [0*, t*], protože předpokládáme nulové hodnoty funkcí pro záporný argument



**Konvoluce, diskrétní aproximace**



kde *O* je lokální okolí “současné pozice”a *h* je konvoluční jádro (též konvoluční maska).

**Příklad**: mějme f = [ 1 2 3 4 5 6 7 8 9 10 11 12 ] - proste nějaká **funkce** a mějme **jádro** třeba h = [ -1 0 1 ]a konvoluce dělá to, že pro každou pozici ve funkci f, vynásobí a sečte, tedy: 4\*(-1) + 5\*0 + 6 \*1 = 2

|  |  |
| --- | --- |
| To je **diskrétní** **2D konvoluce**  Posouvám h a pro každou pozici, pronásobím čtverečky z h s x a všechny je posčítám a uložím do jednoho čtverečku.  Používá se to k rozmazávání (např. když **jádro** je **gausián**), nebo k derivování obrázku (nalezení hran), k nalezení harrisových bodů apod. |  |

**Úloha detekce hran (segmentace obrazu)**

**Hrana:** náhlá změna obrazové funkce

Aplikace hranových detektorů = hornopropustných filtrů

|  |  |
| --- | --- |
| Použiješ ten Prewitt filtr a dostaneš hrany podle y,je to v podstatě derivace podle y.  máš třeba obrázek:  0 0 0  0.5 0.5 0.5  1 1 1  takže je tam přechod z černé do bíle (nahoře černá, dole bílá), dáš na to konvoluci s tím jádrem a dostaneš 0\*1 + 0.5\*0 + (-1)\*1 = -1 (toto 3x podle celé masky a dostaneme -3)|-3| je docela vysoké číslo vzhledem k nule, takže tam **bude** **hrana**, když máme šedý obrázek bez hran:  0.5 0.5 0.5  0.5 0.5 0.5  0.5 0.5 0.5  a uděláme konvoluci, tak dostaneme: 0.5\*1 + 0.5\*0 + 0.5\*(-1) = 0takže tam **hrana** **nebude** |  |

**Úloha úprava obrazu (potlačení šumu)**

Potlačení vysokofrekvenčních složek (šumu) aplikací dolnopropustného filtru.

**Úloha rekonstrukce poškozeného obrazu (ostření)**

Inverzní filtrace nebo Wienerova filtrace

Typ poškození: rozostření pohybem objektu (objektivu), špatné zaostření objektivu, turbulence atmosféry, atd.

**Úloha detekce polohy objektu (detekce objektů)**

|  |  |
| --- | --- |
| Výpočet korelační funkce (k porovnání dvou jevů) = konvoluce v prostor. oblasti  Zrychlení výpočtu aplikací algoritmu FFT ve 2D -> součin ve frekvenční oblasti |  |

**Korelace**

|  |  |
| --- | --- |
| Obecně korelace říká, jak dva jevy spolu souvisí, když je korelace 1, tak hodné souvisí, když 0 tak vůbec nesouvisí, když -1, tak souvisí obráceně.  Na obrázku je vidět hodnota korelace pru různé grafy,  na ose x je třeba **výška** člověka na ose y je třeba **hmotnost** člověka a korelace mezi tím bude třeba 0,8 - takže trochu to spolu souvisí. |  |

Korelace pro určení **podobnosti obrázků**: spočte se v okénku průměrný jas a od všech pixelů v okénku se odečte tato průměrná hodnota a vypočteš směrodatnou odchylku. Součin normalizovaných odchylek je ta korelace, v podstatě to odpovídá cosinu mezi vektory směrodatných odchylek, když mají dva vektory stejný směr, tak úhel je 0, proto korelace je cos 0 = 1, jevy **souvisí**, když jsou vektory směrodatných odchylek na sebe kolmé, tak úhel je 90° a cos 90 = 0, korelace je 0, jevy **nesouvisí**.

**Poznámka** k euklidovské vzdálenosti, podobnost můžeš měřit i eukleidovskou vzdáleností, ne úhlem. Jenže co je tedy lepší? Korelace je invariantní vůči lineárním transformacím jasu, když vypočtu podobnost obrázku (světlého) a ztmaveného obrázku, tak to bude pořád 1, že jsou shodné, ale u eukleidovské vzdálenosti bys dostal, že jsou od sebe nějak vzdáleny, tedy ne úplně shodné.

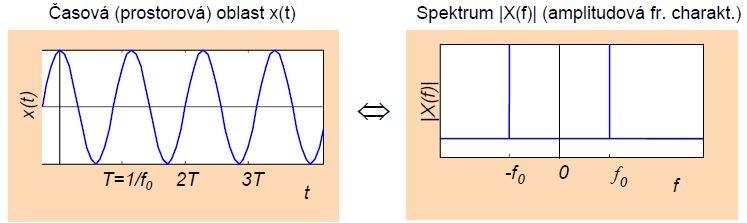
**Rozptyl** je definován jako [střední hodnota](http://cs.wikipedia.org/wiki/St%C5%99edn%C3%AD_hodnota) kvadrátů odchylek od [střední hodnoty](http://cs.wikipedia.org/wiki/St%C5%99edn%C3%AD_hodnota) (od každé hodnoty odečteš průměr, dáš na druhou a z toho spočítáš průměr). **Směrodatná odchylka**, značená [řeckým písmenem](http://cs.wikipedia.org/wiki/%C5%98eck%C3%A1_abeceda) [σ](http://cs.wikipedia.org/wiki/Sigma), se obvykle definuje jako [odmocnina](http://cs.wikipedia.org/wiki/Odmocnina) z [**rozptylu**](http://cs.wikipedia.org/wiki/Rozptyl_%28statistika%29) [náhodné veličiny](http://cs.wikipedia.org/wiki/N%C3%A1hodn%C3%A1_veli%C4%8Dina) X.

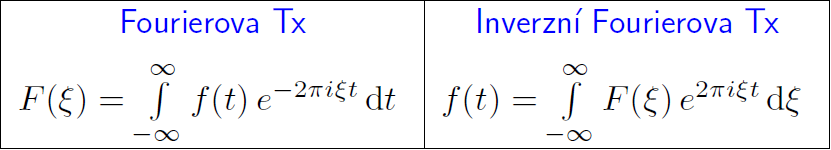
**Fourierova transformace**

**Přímá a inverzní jednorozměrná Fourierova transformace**

Fourierova transformace je vyjádření časově závislého signálu pomocí harmonických signálů, tj. funkcí sin a cos, obecně tedy funkce komplexní exponenciály. Slouží pro převod signálů z časové oblasti do oblasti frekvenční. Signál může být buď ve **spojitém** či **diskrétním** čase.

**Poznámka**: je signál - jednou nahoru, jednou dolu, jednou nahoru, jednou dolů, mění se v čase a ty řekneš, je tam jedna sinusovka o takové frekvenci a to je ten převod. Nebo řekneš je tam jedna sinusovka o takové frekvenci + jiná sinusovka o této frekvenci a popíšeš ten periodicky se opakující signál.



****

Každá 1D funkce se dá vyjádřit jako vážený součet (integrál) mnoha komplexních exponenciál (a díky Eulerově vztahu také jako suma sinusovek a kosinusovek).

**Využití**: ke **kompresi**. Máš nějaký signál - třeba ten obdélnikový a chceš ho zkomprimovat, tak si řekneš, použiji jednu harmonickou (např. sinus) a ten signál zkusím popsat pomocí té sinusovky, bude se to tomu podobat. Čím budeš nasčítávat takových harmonických funkcí víc, tím se bude ten signál víc blížit tomu očekávanému - obdélníkům

no a ty řekněme u JPEGu nastavíš kvalitu třeba 80%, tak ten program si řekne, tak bude mi stačit uložit jen třeba 10 takových harmonických, zbytek informací se zahodí. A pamatuješ si méně informací a signál je skoro stejný jako ten původní obdélníkový. V extrémní kompresi, by sis zapamatoval, třeba jen ten první sinus.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Poznámka:**

****

to je ta fourierova transformace, libovolnou funkci lze přepsat jako součet těchto sinů a kosinů vynásobených nějakými koeficienty. Takže cílem transformace je najít tyhle koeficienty, čím víc koeficientů si pamatuju, tím je to přesnější. No a sin + cos se dá nějak převést jako e^něco\*i, proto se v přednáškách většinou setkáš s tím exponenciálním tvarem a to je vlastně ten eulerův vztah.

**Asymptotická výpočetní složitost jednorozměrné Fourierovy transformace** je **O**(N^2).

Fourierova transformace předpokládá **periodický** (opakující) **signál**. A co když potřebujeme zpracovávat neperiodický signál? Dva obvyklé přístupy.

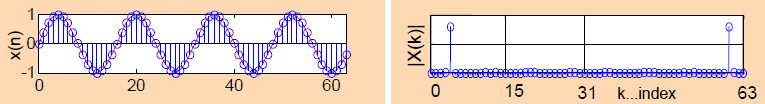
1. Zpracovat signál po malých částech (oknech) a předpokládat, že vně je signál periodický.

2. Použití složitějších bázových funkcí, např. vlnek ve vlnkové (wavelets) transformaci.

**Diskrétní Fourierova transformace**

Uvažujme vstupní signál (posloupnost) *f*(*n*), *n* = 0*, . . . ,N −* 1.

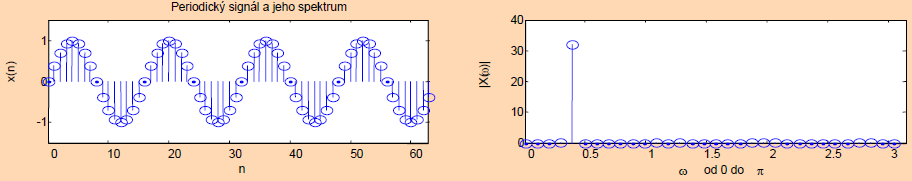
|  |  |
| --- | --- |
|  |  |



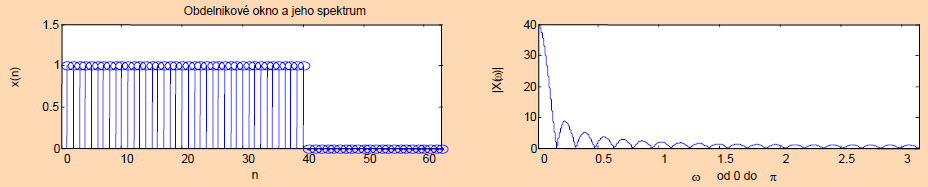
**Poznámka**: navzorkujeme ten signál a udělá se transformace, vlevo je vzorec pro výpočet funkce X(k) obdoba F(k) a průběh té funkce X(k) je znázorněn vpravo na obrázku. A existuje inverzní transformace, jejíž vzorec vidíš vpravo. A když ho použiješ na tu funkci X(k), tak dostaneš zase zpět tu funkci x(k), která je na obrázku vlevo.

Příklad: Vpravo je spektrum, tam nejsou periody - žádné vlnové délky, vlevo je periodická funkce, vpravo ti to jen říká kolik je kterých frekvencí.

Frekvence je počet kmitů za sekundu, perioda je pak kolik trvá jeden kmit.



Je tam jen jedna sinusovka o nějaké frekvenci blízko f = 0.5.



Nasčítá se hodně sinusovek a čím vyšší frekvence, tím méně bude přispívat k součtu, což je logické, protože je tam jedna sinusovka s hodně velkou vlnovou délkou, která je shodná s tím obdélníkem, tedy je s nízkou frekvencí a pak ty s vyšší frekvenci, už se uplatňují méně. Kdybych je uplatnil více, tak bych si to rozhodil ..jinak, vysoké frekvence obvykle odpovídají šumu.

**Rychlá Fourierova transformace (*FFT – fast Fourier transform*)**

FFT má výpocetní složitost **: **

FFT je efektivní algoritmus pro spočtení diskrétní Fourierovy transformace (DFT) a její inverze. FFT je velmi důležitá v mnoha oblastech, od digitálního zpracování signálu a řešení parciálních diferenciálních rovnic až po rychlé násobení velkých celých čísel.

Myšlenka FFT: posloupnost délky N lze vyjádřit jako součet dvou posloupností o délce N/2. V jedné jsou **liché** a v druhé **sudé** vzorky. Provádí se rekurzivní výpočet, N musí být mocninou 2. Kromě toho se využívá i motýlkové schéma výpočtu - znovuvyužití vypočtů.

|  |  |
| --- | --- |
| **Motýlkový algoritmus** Samotné skládání jednotlivých podposloupností provádí specializovaný algoritmus tzv. Motýlkový algoritmus. Ten provádí postupné skládání prvků z **liché** a **sudé** posloupnosti do dvou po sobě jdoucí prvků nové posloupnosti.  **Omezení pro vstupní obraz** Obrázek musí být před FFT transformován tak aby počet řádků a sloupců byl mocninou 2. A to například „scalováním“ obrázku do nejbližší mocniny 2. |  |

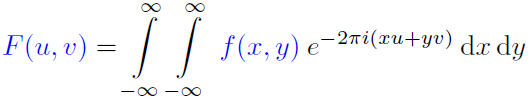
**2D Fourierova transformace**

Fourierova transformace lze zevšeobecnit do vyšších dimenzí. Například, mnoho signálů jsou funkce 2D prostoru vymezeného nad rovinou xy. 2-dimensionalní Fourierova transformace má také čtyři různé formy v závislosti na tom, zda 2D signálu je periodické a diskrétní.

***Myšlenka****.* Obrazová funkce *f*(*x, y*) se rozloží na lineární kombinaci harmonických (sínusovek, kosínusovek, obecneji ortonormálních) funkcí.

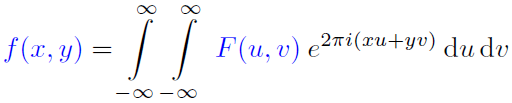
Dvojrozměrná Fourierova transformace umožňuje převést rozložení obrazových intenzit f(x, y) vyhodnocovaného obrazu na obraz prostorových frekvencí F(u, v).

**Definice přímé transformace**. u, v jsou prostorové frekvence.



2D FT se používá při zpracování obrazů, filtraci, rekonstrukci, kompresi atd…

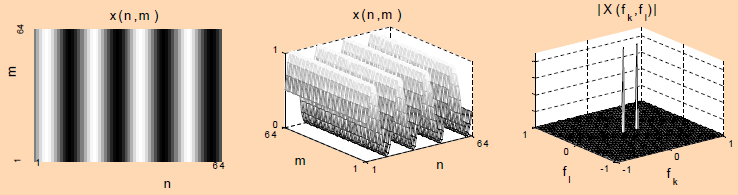
**Inverzní Fourierova transformace**



***f*(*x, y*)** je lineární kombinací jednoduchých harmonických složek 

Díky Eulerovu vztahu () jsou **reálnými** složkami cos a **imaginárními** sin.

Funkce ***F*(*u, v*)** (komplexní spektrum) udává váhy harmonických složek v lineární kombinaci.



**Poznámka**: sinus je od -1 do +1, jenže obrázek vlevo je od 0 - černá do 1 bílá, takže tam musí být šedé (0,5) pozadí a budu od toho jednou něco odečítat a půjdu k černé (0) a jednou něco přičítat a půjdu k bílé (1).

0.5 + 0.5\*sin(x) prvních 0.5 odpovídá tomu prvnímu vrcholu (0,0) a ten druhý vrchol tomu druhému vrcholu (obr. napravo), tedy druhých 0,5 druhému vrcholu. Tímhle se to pravě převádí do obrázku (i normalizuje), prostě konstantní signál je také periodický (s nulovou amplitudou).

V tom šedém rámečku (nahoře) máš a0, což je vlastně to samé, k tomu a0 jenom přičítají, nic neodečítají. Jasně, ale co dělá sinus? Nabývá hodnot od 1 po -1 občas.

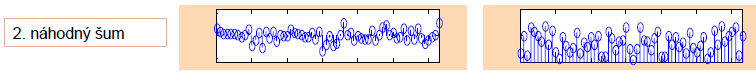
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

viz. <http://www.jcrystal.com/steffenweber/JAVA/jfourier/jfourier.html>

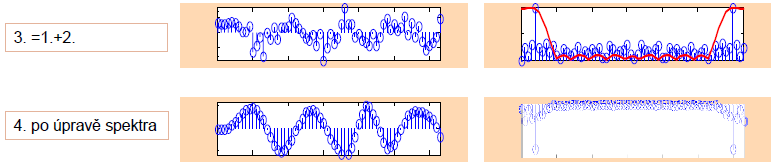
Výška vrcholu (v transformaci) je v podstatě amplituda sinusovky (v obrázku nebo nějakém grafu).



Když je signál periodický, tak stačí dva vrcholy pro 2D ve spektru.



Ale v případě šumu je tedy třeba použít spoustu sinusovek a cosinusovek. Šumy jsou většinou v obrázku malé tečky, což odpovídá většinou vysokým frekvencím .. protože velké věci mají nízké frekvence.



Přidali šum k prvnímu a v posledním šum odstranili použitím konvoluce nějakým gausiánem, ten to rozmaže a ztratí se šum.

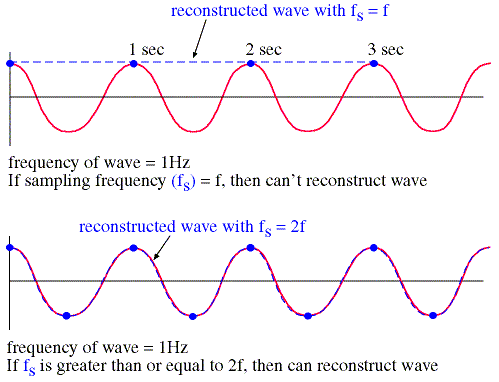
**Shannonova (též Nyquistova, Kotelnikova) větu o vzorkování**

Přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň **dvakrát vyšší** než je **maximální frekvence rekonstruovaného signálu**.

V praxi se tedy vzorkovací frekvence volí dvakrát větší plus ještě nějaká **rezerva** než je maximální požadovaná přenášená frekvence. V telekomunikacích je to např. 8 kHz neboť je třeba přenášet pouze signály ve standardním telefonním pásmu (od 0,3 do 3,4 kHz zaokrouhleno směrem nahoru 4 kHz). Například u záznamu na CD je to zas 44,1 kHz neboť zdravé lidské ucho slyší maximálně cca do 20 kHz a tudíž vzorkovací frekvence 44,1 kHz byla zvolena s velkou rezervou.

V případě použití nižší vzorkovací frekvence může dojít k tzv. **aliasingu**, kdy rekonstruovaný signál je výrazně odlišný od původního vzorkovaného signálu.

Dokazuje je to například tento obrázek, kde je v prvním případě vzorkovací frekvence nižší.



**Vzorkování obrazu**

|  |  |
| --- | --- |
| Zahrnuje dvě úlohy:  1. Uspořádání vzorkovacích bodů do rastru.  2. Vzdálenost mezi vzorky (Nyquist-Shannonova věta o vzorkování).  Vzorkovací frekvence > 2 krát maximální frekvence v signálu (tj. nejvyšší frekvence zcela rekonstruovatelná z vzorkovaného signálu).  V obrazech musí velikost vzorku (pixelu) být dvakrát menší než nejmenší detail, který chceme zaznamenat. |  |

**Rekonstrukce obrazu z navzorkovaných dat**

Úkolem je zpětná rekonstrukce obrazu, tj. zvětšení rozlišení obrázku z 64x64 pixelů na 256x256 pixelů. Protože je nový obraz 4x větší, musíme určit hodnoty jasu v chybějících bodech. To provedeme tak, že aproximujeme průběh plochy výstupního obrazu z výše uvedených vzorků a stanovíme hodnoty jasů v novém rastru. Provedeme to tak, že vzorky proložíme polynomem. Přesnost aporoximace ovlivňuje její postup a počet vzorků.

Matlab nám nabízí několik způsobů aproximace:

* **Metoda nejbližšího souseda** - přiřadí zpracovávanému bodu hodnotu jasu nejbližšího souseda.
* **Lineární interpolace** - pro aproximaci se využijí 4 sousední body v okolí zpracovávaného bodu. Výsledná obrazová funkce je lineární kombinací každého ze čtyř bodů a navíc je úměrná vzdálenosti od zpracovávaného bodu.
* **Bikubická interpolace** - pro aproximaci se používá okolí složené ze 16-ti bodů. Model obrazové funkce je zpřesněn tím, že je lokálně interpolován bikubickým polynomem.
* **Kubická interpolace** - na rozdíl od předchozích způsobů je zde využita trojúhelníková síť a pro aproximaci se používá 10-ti bodů. Model obrazové funkce je lokálně interpolován kubickým polynomem. Oproti bikubické interpolaci má kubická tu výhodu, že aproximuje polynomem nižšího řádu. Aproximační funkce má tedy větší stabilitu.

Pro zpětnou rekonstrukci obrazu jsme použili vzorek, kde jsme pro potlačení vysokých frekvencí použili FFT a filtru s Gaussovým rozložením (sigma=12).

Protože se odfiltrováním vysokých frekvencí z obrazu se ztratilo velké množství informace, neodpovídá zpětně rekonstruovaný obraz obrazu původnímu. Jako nejlepší metoda pro rekonstrukci obrazu se jeví interpolace pomocí kubického polynomu.

|  |  |  |
| --- | --- | --- |
| Původní obrázek | Nejbližší soused | Kubická interpolace |

**Interpolace obrazu**

Interpolace je proložení neznámých hodnot mezi dvěma "poly", třeba máš malý obrázek a ten zvětšíš, tak to musíš něčím proložit, jednotlivé barevné body se natáhnou a mezi nimi není nic (prázdno), potřebuješ ten prostor mezi body něčím vyplnit, tak počítáš interpolaci. Tedy pro místo mezi dvěma body vypočteš jas.

Po geometrické transformaci nemusí vycházet body v místě odpovídajícím mřížce nového obrázku (jinak řečeno, po transformaci vyjdou souřadnice v reálných číslech, ale mřížku máme v celých číslech), poté k vyjádření hodnoty jasu v místě využíváme různé interpolační metody (popsány jen dvě):

* **nejbližší soused** - přiřadí bodu (x, y) hodnotu jasu nejbližšího bodu gs v diskrétní mřížce (najde nejbližší bod a použije stejnou hodnotu jasu)
* **linearní interpolace** - využije 4 body v okolí (x, y) a lineárně je zkombinuje. Vliv každého ze čtyř bodů v lineární kombinaci je úměrný jeho blízkosti ke zpracovávanému bodu.

Nebo můžeš mít interpolaci barev, máš **červený** bod na souřadnicích (0,0) a **modrý** bod na souřadnicích třeba (1,1)

co bude v bodě (0.5, 0.5) ? nebo jinak, co bude v bodě (0.8, 0.8) ? podle nejbližšího souseda tam dám modrou, podle lineární interpolace tam dám víc modré a méně červené podle vzdálenosti od sousedních bodů.